

## Efficient geocasting in opportunistic networks

Article (Accepted Version)

Rajaei, Aydin, Chalmers, Dan, Wakeman, Ian and Parisi, George (2018) Efficient geocasting in opportunistic networks. Computer Communications, 127. pp. 105-121. ISSN 0140-3664

This version is available from Sussex Research Online: <http://sro.sussex.ac.uk/id/eprint/76149/>

This document is made available in accordance with publisher policies and may differ from the published version or from the version of record. If you wish to cite this item you are advised to consult the publisher's version. Please see the URL above for details on accessing the published version.

### **Copyright and reuse:**

Sussex Research Online is a digital repository of the research output of the University.

Copyright and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable, the material made available in SRO has been checked for eligibility before being made available.

Copies of full text items generally can be reproduced, displayed or performed and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

Efficient Geocasting in Opportunistic Networks

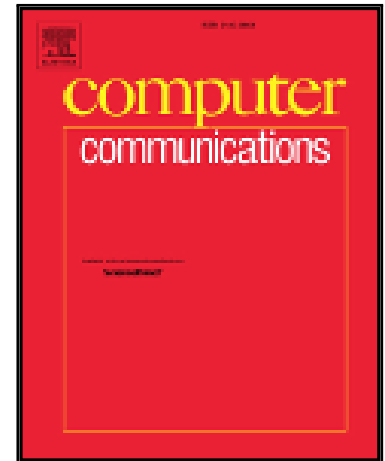
Aydin Rajaei, Dan Chalmers, Ian Wakeman, George Parisis

PII: S0140-3664(17)31315-4  
DOI: [10.1016/j.comcom.2018.05.014](https://doi.org/10.1016/j.comcom.2018.05.014)  
Reference: COMCOM 5707

To appear in: *Computer Communications*

Received date: 30 December 2017  
Revised date: 30 March 2018  
Accepted date: 21 May 2018

Please cite this article as: Aydin Rajaei, Dan Chalmers, Ian Wakeman, George Parisis, Efficient Geocasting in Opportunistic Networks, *Computer Communications* (2018), doi: [10.1016/j.comcom.2018.05.014](https://doi.org/10.1016/j.comcom.2018.05.014)



This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

# Efficient Geocasting in Opportunistic Networks

Aydin Rajaei, Dan Chalmers, Ian Wakeman, and George Parisi\*

*School of Engineering and Informatics, University of Sussex, UK.*

## Abstract

With the proliferation of smartphones and their advanced connectivity capabilities, opportunistic networks have gained a lot of traction during the past years; they are suitable for increasing network capacity and sharing ephemeral, localised content. They can also offload traffic from cellular networks to device-to-device ones, when cellular networks are heavily stressed. Opportunistic networks can play a crucial role in communication scenarios where the network infrastructure is inaccessible due to natural disasters, large-scale terrorist attacks or government censorship. Geocasting, where messages are destined to specific locations (casts) instead of explicitly identified devices, has a large potential in real world opportunistic networks, however it has attracted little attention in the context of opportunistic networking.

In this paper we propose *Geocasting Spray And Flood (GSAF)*, a simple and efficient geocasting protocol for opportunistic networks. GSAF follows an elegant and flexible approach where messages take random walks towards the destination cast. Messages that are routed away from the destination cast are extinct when devices' buffers get full, freeing space for new messages to be delivered. In GSAF, casts do not have to be pre-defined; instead users can route messages to arbitrarily defined casts. GSAF does that in a privacy-preserving fashion. We also present *DA-GSAF*, a *Direction-Aware extension of GSAF* in

---

\*Aydin Rajaei, Ian Wakeman and George Parisi are with the School of Engineering and Informatics, University of Sussex, UK.

E-mail: {A.Rajaei, I.J.Wakeman, G.Parisi}@sussex.ac.uk

Dan Chalmers is an Honorary Senior Research Fellow in the School of Engineering and Informatics, University of Sussex, UK.

E-mail: D.Chalmers@sussex.ac.uk

which messages are forwarded to encountered nodes based on whether a node is moving towards their destination cast. In DA-GSAF only the direction of a mobile node is revealed to other devices. We experimentally evaluate our protocols and compare their performance to prominent geocasting protocols in a very wide set of scenarios, including different maps, mobility models and user populations. Both GSAF and DA-GSAF perform significantly better compared to all other studied protocols, in terms of message delivery ratio, latency and network overhead. DA-GSAF is particularly efficient in sparse scenarios minimising network overhead compared to all other studied protocols. Both GSAF and DA-GSAF perform very well for a wide range of device/user populations indicating that our proposal is viable for crowded and sparse opportunistic networks.

*Keywords:* Geocasting, Opportunistic Networks, OppNets, Delay-Tolerant Networks, DTNs, Opportunistic Routing.

---

## 1. Introduction

The proliferation of smartphones and their long- and short-range connectivity capabilities have made the deployment of opportunistic networks [1][2] a realistic [3] and viable solution to a number of problems. Wireless technologies, such as LTE, Wi-Fi, Wi-Fi Direct and Bluetooth, allow smartphones to access the Internet as well as communicate with devices within their range, in an ad-hoc, peer-to-peer fashion [3][4]. Opportunistic networks have gained a lot of traction during the past years; they are suitable for increasing network capacity [5][6] and sharing ephemeral, localised content [7]. They are also appropriate for offloading traffic from cellular networks to device-to-device ones, whose formation is assisted by cellular providers [8][9], who have strong incentives to do so when their networks are heavily stressed [10][11]. Equally importantly, opportunistic networks can play a crucial role in communication scenarios where the network infrastructure is (partially or fully) inaccessible due to natural disasters, large-scale terrorist attacks or government censorship. They can also be the means for (localised) communication when the network infrastructure is not

trusted. For example, FireChat<sup>1</sup> has been extensively used during the protests in Hong Kong in 2014<sup>2</sup>.

In most of the scenarios described above, communication and content dissemination is geographically confined (e.g. within a city or a region where a natural disaster took place or a part of the city where protesters demonstrate). Apart from being able to send messages to a specific device in the network (unicasting), it is also crucial to be able to route messages to specific geographical locations (*geocasting*) within the opportunistic network. Effective geocasting has a large potential in the real world use of opportunistic networks: (1) geographical notifications for emergency situations, such as fire and natural disasters; (2) location targeted advertising, where a large volume of users is concentrated at specific locations (e.g. open festival venues or large stadiums) to attend music festivals, sports events or to participate in a demonstration; (3) geographically restricted service discovery. These geographical locations (*casts*) may be pre-defined, even before a network is deployed, or specified by the sender for each message, separately. A temporal aspect is also relevant to geocasting, apart from the spatial one; destination nodes must receive a message before it expires; e.g. before a notification or an evacuation instruction becomes invalid in a natural disaster scenario.

Unicasting has been extensively studied in the context of Opportunistic networks [12] [13], but none of the existing protocols can support geocasting, given that unicast protocols route messages to specific devices, which are explicitly identified by unique endpoint identifiers. A number of geographical routing protocols that utilise the location of network devices to efficiently route messages in opportunistic networks have been proposed [14]. Note that these protocols are unicast protocols and are not designed for geocasting. Geocasting has been mostly studied in the context of Mobile Ad Hoc Networks (MANETs) [15].

<sup>1</sup><http://tinyurl.com/ogsz75o>

<sup>2</sup>500,000 downloaded the application in Hong Kong alone during the first two weeks of the protests.

MANETs present radically different properties compared to opportunistic networks. In MANETs, connectivity (as well as the overall network topology) between mobile nodes is rather stable; no such assumptions can be made for opportunistic networks, where mobility is high<sup>3</sup> and connectivity is very intermittent. As a result, no end-to-end paths among all nodes exist at all times and the network topology is unknown and constantly changing. Hence, none of the existing geocasting protocols for MANETs are suitable for opportunistic networks.

In this paper we present *Geocasting Spray And Flood (GSAF)* and its Direction-Aware extension (DA-GSAF); they are both simple but efficient and flexible geocasting protocols for opportunistic networks, which overcome limitations of existing approaches. Contrary to protocols where casts must be pre-defined [7] [16], or defined as circles (by defining a centre point and a radius) [7] [17] [18] [19], or the network should be divided into pre-defined, non-overlapped equal cells [17] [18] [19], our approach allows for flexible definition of casts as a polygon defined by set of coordinates. The sender defines the cast, the cast definition is carried in the routed message and other nodes only check whether they reside within the defined cast. This flexibility is required in many scenarios where fine-grained specification of casts is crucial (e.g. for fine-grained emergency notifications to avoid widespread panic). Moreover, in our approach a device can send a message in a cast even if it does not reside in it. This is in contrast to [7], where devices can only publish content within the region they reside. With such an approach, it would be very inefficient to reach relatively remote regions by just increasing the radius of the cast, effectively flooding a very large area with, probably, unwanted content.

In our approach, devices do not exchange any location-related information, thus preserving users' privacy, and take routing decisions autonomously. This is in contrast to approaches that exchange explicit [18] or aggregated location

---

<sup>3</sup>Mobility is actually exploited so that messages are physically carried in the devices towards their final destinations.

information (e.g. cast visiting probabilities [17]) or information that is used to collaboratively build mobility maps [16]. Exchanging location information requires network bandwidth and battery, resources that are precious, and rather  
 75 scarce in opportunistic networks. Expensive computations (e.g. as in [16][17]) also drain the battery quickly. In [18] the network is partitioned into two layers, requiring either a third party to perform the partitioning or a distributed consensus protocol for electing nodes to be in each of these layers (consuming bandwidth especially under high node churn).

80 GSAF follows a simple but effective approach where messages take random walks towards the destination cast. Messages that follow directions away from the cast are deleted when device buffers get full, freeing space for new messages to be delivered. In brief, message dissemination is as follows: upon receiving a message, a node checks whether it is a destination node (the definition of  
 85 destination node(s) incorporates both spatial and temporal aspects, as described in Section 2) or not. This requires devices' location services and presents a well-known trade-off with respect to the accuracy of the reported location (which, in turn, affects the granularity of cast definition) and battery consumption. In the latter case, a device carries and forwards the message to other nodes based  
 90 on a ticketing mechanism, inspired from [20]. When a message reaches a cast, it is disseminated through controlled flooding and does not get re-routed if it exits the destination cast. Expired messages are discarded. GSAF is a privacy-preserving protocol.

DA-GSAF extends GSAF by adding direction (but not destination) aware-  
 95 ness in the way routing decisions are taken. More specifically, during its first phase, preference is given to devices that head towards the direction of a message's destination cast. DA-GSAF is not strictly privacy-preserving, since it requires devices to reveal their direction to other devices before exchanging messages, albeit in a coarse-grained fashion. This is in contrast to approaches  
 100 like [16] [17] [18]. As shown in Section 4, DA-GSAF performs the best with respect to message delivery ratio and network overhead in scenarios with small numbers of users and sparser casts.

In Section 4 we present an extensive evaluation of the proposed geocasting protocols. We have implemented GSAF and DA-GSAF, as well as the most prominent geocasting protocols (i.e. [17], [18] and [21]) in the ONE simulator and have experimented with different maps (University of Sussex Campus and Helsinki's City Centre), mobility models (working day, random), numbers of mobile users and wireless communication technologies (WiFi and Bluetooth). We have compared the performance of all implemented protocols in terms of message delivery ratio, latency and network overhead. Our protocols significantly outperform all other considered protocols, in all simulated scenarios. Our scalability study shows that both GSAF and DA-GSAF perform well in scenarios where the number of mobile users is large.

## 2. Geocasting in Opportunistic Networks

Before proceeding with the detailed description of the proposed geocasting protocols, we discuss challenges that influenced our work. Geocasting in opportunistic networks entails both spatial and temporal aspects and needs to take into account both network and device resource constraints.

**Objectives.** In geocasting the goal is to successfully deliver a message to all users (or to as many as possible) that reside inside a geographical area within a specific time interval. It is not only necessary for a message to reach a cast, but it must also be efficiently disseminated within the cast. The temporal aspect is crucial because, in many communication scenarios, messages may be invalidated or deleted from the network, either because the information they carry expires or just because there can be no guarantees that a message will reside within a cast indefinitely. Messages can become valid after their creation and initial forwarding. This feature is important in scenarios where messages are time-sensitive and only valid for a specific time duration. For example, a sender could pro-actively generate and send messages to casts and these messages become valid (and therefore deliverable to end-user applications) at a later time; e.g. in time-sensitive, geographically-specific advertisements or disaster scenarios for



specific notifications or evacuation instructions.

Both the spatial and temporal aspects are relevant in defining a destination node of a message in our communication scenario; (1) destination nodes must reside in the message's destination cast when receiving the message and (2) the time at which they receive the message must fall within the defined lifetime that is carried in it. Messages whose lifetime has expired are discarded. A node that receives a message within its destination cast, whose lifetime has not started yet will forward the message as normal. It may also store the message locally and make it available to an interested application when the lifetime starts. Messages in geocasting must carry information about their destination, as their *Endpoint Identifier (EID)*. For example, if casts are pre-defined at deployment time and known to all devices, a message may carry a cast identifier; otherwise, the cast definition (e.g. centre/radius pair or coordinates of a polygon, as in our approach) must be carried in the message. Whenever a node receives a message, it compares its own location with the EID of the message.

**User and Device Characteristics.** In opportunistic networks, mobile devices support location services, which may vary in the supported accuracy (and the associated battery consumption). Access to GPS for outdoors scenarios is ideal, although in most cases, coarser-grained estimations are sufficient. For indoors scenarios, relevant localisation approaches (e.g [22]) can be used. Network density in terms of mobile devices roaming within an opportunistic network may vary significantly for different scenarios but also in time. Flooding the network may work well in very sparse scenarios, although the network overhead would be significantly increased as the number of users increases. Accordingly, a protocol that forwards packets very selectively (e.g. by calculating cast visiting probabilities [17]) may result in low network overhead in dense scenarios but very low message delivery ratio in sparse scenarios. In any case, the ideal geocasting protocol should perform well in both sparse and dense scenarios.

**Resources and Constraints.** Opportunistic networks employ *store-carry-forward* based mechanism for message routing (including geocasting), therefore

mobile devices must be able to temporarily store and carry messages before they forward them to other devices. Although devices' memory has grown over the past years, one would not expect to be able to utilise more than some  
 165 tens of MBs of memory in each device, given that other applications and background services require access to ever increasing chunks of memory. This has implications in the way data is forwarded. For example, unconstrained message flooding would result to quickly filling up devices' buffers, resulting in the fast extinction of messages in the network. Increasing the size of available buffers  
 170 in each device does not simply solve the problem described above. Network bandwidth is limited but most importantly the time period that two devices can exchange messages is short, given that users move. As a result, if very large buffers were used, only a very small portion of the carried messages could be forwarded from device to device. Forwarding also comes with a cost in terms  
 175 of battery consumption. Control-messages exchanged among devices (e.g. to build mobility maps [16]) as well as local computations of metrics that influence how routing is done (e.g. as in [17]) may result in quick battery drain.

**User Privacy.** Users are very concerned when it comes to giving away their privacy in terms of mobility patterns, future destinations or social interactions  
 180 for the sake of a more efficient routing protocol [23]. Exchanging location-related information among devices has significant privacy implications that must be taken into consideration when designing a geocasting protocol for opportunistic networks. Ideally, a geocasting protocol should be effective and efficient without requiring users to release any information that may be considered private.

### 185 3. Design

In this section we present GSAF and DA-GSAF and explain how messages are routed to their destination casts using a two-phased approach.

#### 3.1. Cast Definition and Membership Check

Geographical casts effectively define a group of users that reside in the same  
 190 region and can be addressed by geocasting messages to this specific cast. In the

following, we describe (1) how casts are defined<sup>4</sup> and (2) how mobile devices check whether they are recipients of a message.

A cast is defined as a set of coordinates in a two-dimensional space (the network). The coordinates define the edges of the cast. Figure 1 depicts an example of a cast definition inside a map. With this approach users can send their messages to arbitrarily defined casts. This is a fine-grained method to define a cast which provides great flexibility, potentially minimising the number of devices that are receiving unwanted messages, compared to other geocasting approaches that define casts as circles (i.e. as centre/radius pairs). With approaches like [7] [17], if a specific region, which is far from the centre of the circle, needs to be reached, the radius has to be increased, resulting in wasted network bandwidth for messages that reach devices for which the message is useless. In our approach, users can draw the destination casts on their mobile phones effectively defining message destinations on-the-fly. Messages carry the defined cast information (the set of coordinates) as their delivery EID.

When a mobile device receives a message, it checks whether it is located inside or outside the destination cast; i.e. whether it is a potential recipient or not. Given that this check is performed for every received message, the algorithm must be very fast. Indeed, a number of very efficient algorithms have been proposed in the past in the context of the *Point-in-Polygon* problem, a well studied problem in computer graphics and image processing [24]. As its name suggests, solutions to this problem check whether a specific point is inside a polygon or not. According to Haines [24], three main techniques can be used to solve this problem; the *crossing test*, *angle summation test* and *triangle test*. Among these, the crossing test is the fastest (as shown in [24]) and therefore has been adopted in our work. An example of the crossing test is illustrated in Figure 2. Initially, a vertical (to the x-axis) line that crosses the point (with coordinates  $(x_p, y_p)$ ) that needs to be checked is drawn. The point  $(x_p, y_p)$  is the

---

<sup>4</sup>Note that casts do not have to be pre-defined. Instead, a sender can define a cast to send a message to, on-the-fly.

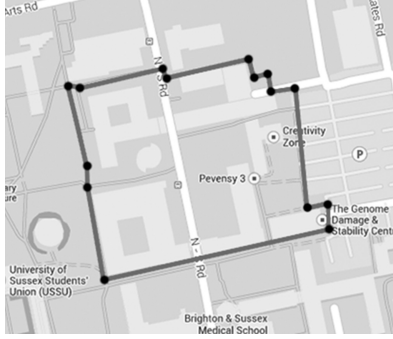


Figure 1: Cast Definition on a Map

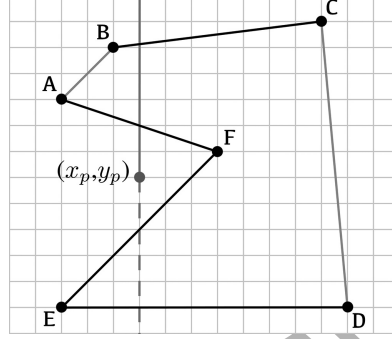
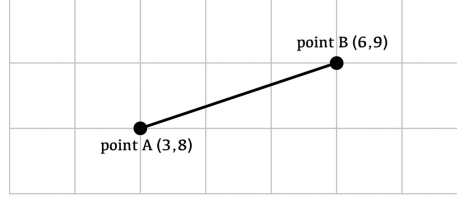


Figure 2: Crossing Test Example

initial point that decomposes this line into two rays (half-lines). The number of  
 intersections of one of the rays (e.g. the vertical solid line in Figure 2) with the  
 sides of the polygon is used to check whether the point is in the polygon or not;  
 if the number is even, the point is located outside the polygon, otherwise the  
 point is inside. For each pair of neighbouring polygon coordinates, we calculate  
 the parameters of the line equation that defines the line that connects these  
 two points, as shown in Figure 3. The  $x_{range}$  defines the projection of each  
 side of the polygon to the x-axis. In order to calculate the intersections of the  
 vertical line with the given polygon, we simply test whether  $x_p$  is within the  
 $x_{range}$  for each side of the polygon. For example, in Figure 2, sides  $BC$ ,  $DE$ ,  $EF$   
 and  $FA$  intersect with the vertical line. Next, we calculate the  $y$  coordinates of  
 the intersection points by solving the line equation that defines each side of the  
 polygon using  $x_p$ . Finally, we count the number of  $y$  coordinates that are larger  
 than  $y_p$  (i.e. looking at the ray illustrated with vertical solid line). This dictates  
 whether the device  $(x_p, y_p)$  is inside (odd number) or outside (even number) the  
 cast. Algorithm 1 implements the crossing test and checks whether a node is  
 located inside a cast or not. In our simulations, the crossing test performs  
 marginally worse compared to checking whether a point is within a circle. We  
 argue though that this slight performance penalty is negligible with respect to  
 routing performance and battery consumption.



Linear equation :

$$y - y_A = \frac{y_B - y_A}{x_B - x_A} (x - x_A) \quad x_{range} : [x_A, x_B] \Rightarrow$$

$$y - 8 = \frac{9 - 8}{6 - 3} (x - 3) \quad x_{range} : [3, 6] \Rightarrow$$

$$y = \frac{1}{3}x + 7 \quad x_{range} : [3, 6]$$

Figure 3: Parameters of the Line Equation

### 3.2. Message Lifetime

240

As mentioned in Section 2, cast membership is dynamic due to the inherent user mobility and therefore defining a time interval during which a message is valid is crucial. Our approach follows the *Current-Member Delivery* membership

---

#### Algorithm 1 Crossing Test

---

Require:

$N$  : the current node.

$C_m$  : the destination cast of message  $m$ .

1: **function** ISINSIDECAST( $N, C_m$ )

Let  $N_{(x,y)}$  be the location  $(x,y)$  of  $N$

Let  $y_{(x_N)}$  be the equation of vertical line which passes through  $x_N$

Let  $pointsList\{P_{(x,y)}\}$  be the list of  $C_m$ 's boundary points  $P_{(x,y)}$  which intersect  $y_{(x_N)}$

2: **for** each point  $P_{(x,y)}$  in the  $pointsList\{P_{(x,y)}\}$  **do**

3: **if** ( $y_P < y_N$ ) **then**

4: **exclude** current  $P_{(x,y)}$  from  $pointsList\{P_{(x,y)}\}$

5: **end if**

6: **end for**

7: **if**  $pointsList.size()$  is an odd number **then**

8: **return** *True* ( $N_{(x,y)}$  is inside  $C_m$ )

9: **else**

10: **return** *False* ( $N_{(x,y)}$  is outside  $C_m$ )

11: **end if**

12: **end function**

---

model, as defined in [25]. Instead of just defining a single lifetime value, we use a pair of epoch times to define the beginning and end of the message's life. With this approach we enable geocasting messages that will become valid in the future to cater for anticipated latencies to reach the destination cast or specific use cases e.g. where an advertisement for sales becomes valid at rush hours in a large shopping mall.

### 3.3. Geocasting Spray And Flood (GSAF)

GSAF employs a two-phased approach to route messages towards their destination cast (phase 1) and spreading them to recipient devices within the destination cast (phase 2). The two-phased approach facilitates simplicity and elegance and is inspired by *Unicast Routing with Area Delivery* (URAD), a geocasting approach for MANETs [15]. *Recipients of a message are the nodes that are present at its destination cast when they receive it and they do so during the lifetime interval defined in it.*

**Phase 1 - Forwarding (and carrying) messages to their destination cast.** In the first phase, GSAF follows a multi-copy spraying approach (inspired by [20]), which is fast in terms of reaching the destination cast as well as efficient in terms of message delivery ratio and network overhead. Algorithm 2 specifies how GSAF routes messages in the network. Upon message creation,  $T$  tickets are “assigned” to it (represented as a ticket counter which is included in the message header).  $T$  denotes the number of times a message can be forwarded to encountered devices from a specific device. Each time a message is copied and forwarded to another node,  $T$  is decreased by 1 in both devices (*Lines 10 and 27 of Algorithm 2*). Note that at this point both devices have  $T - 1$  copies and therefore they can independently pass the message to  $T - 1$  devices each. When  $T$  gets to zero, the message cannot be forwarded any further. It will be deleted when the local buffer gets full or when the message expires. Until then it can only be carried by the device (maybe until it physically reaches the destination cast). *Note that before forwarding messages upon encountering another device, a node will first check which messages it shares with the neighbouring node so*

that messages are not exchanged unnecessarily, minimising the required network overhead. This is done by first exchanging message digests, as defined in Lines

---

**Algorithm 2** GSAF Routing
 

---

Let  $C_m$  be the destination cast of message  $m$   
 Let  $T_m$  be the # of available tickets for message  $m$   
 Let  $B_x$  be the set of messages in node  $N_x$  buffer  
 Let  $D(B)$  be the set of message digests from buffer  $B$

```

1: for each encounter with  $N_j$  do
2:   As a sender do
3:   drop expired messages from  $B_i$ 
4:   apply buffer scheduling policy
5:   send  $D(B_i)$  to  $N_j$ 
6:   wait for  $D(B_i \cap B_j)$ 
7:   for each message  $m$  in  $B_i \setminus B_j$  do
8:     if ( $T_m > 0$ ) then
9:       forward  $m$  to  $N_j$ 
10:       $T_m \leftarrow T_m - 1$ 
11:     else if ( $T_m = 0$ ) then
12:       if  $isInsideCast(N_i, C_m)$  (Algorithm 1) then
13:         forward  $m$  to  $N_j$ 
14:       end if
15:     end if
16:   end for
17:   As a receiver do
18:   wait for  $D(B_j)$  from  $N_j$ 
19:   send  $D(B_i \cap B_j)$ 
20:   for each received message  $m$  do
21:     if  $isInsideCast(N_i, C_m)$  (Algorithm 1) then
22:       deliver  $m$  to the Application Layer
23:       add  $m$  into  $B_i$ 
24:        $T_m \leftarrow 0$ 
25:     else
26:       add  $m$  into  $B_i$ 
27:        $T_m \leftarrow T_m - 1$ 
28:     end if
29:   end for
30: end for
  
```

---

275 5 and 6.

**Phase 2 - Delivering messages to recipient devices inside their destination cast.** In the second phase, a message is disseminated to all devices inside its destination cast by following an intelligent flooding approach. GSAF floods the message to nodes inside the cast by handing a copy to them (Lines 12 to 14 and 21 to 24 of Algorithm 2). If a copy of the message goes out of the destination cast, it can only be forwarded back to nodes inside the cast. The message will sit in the device's buffer until it expires and gets deleted (Line 3 of Algorithm 2). This way, unnecessary message exchanges outside the cast are prevented.

285 Figure 4 illustrates an example of message dissemination. The sender creates a new message and initialises  $T$  to 3. It encounters two nodes (one after the other) and for each such encounter, it decreases the value of  $T$  in the message and forwards a copy of the message to the encountered node. As shown in the figure,  $T$  is first decreased to 2 (which is also the value of  $T$  in the message held by the node above the source node) and, then, to 1 (the value of  $T$  in the message received by the node below the source node). The same takes place

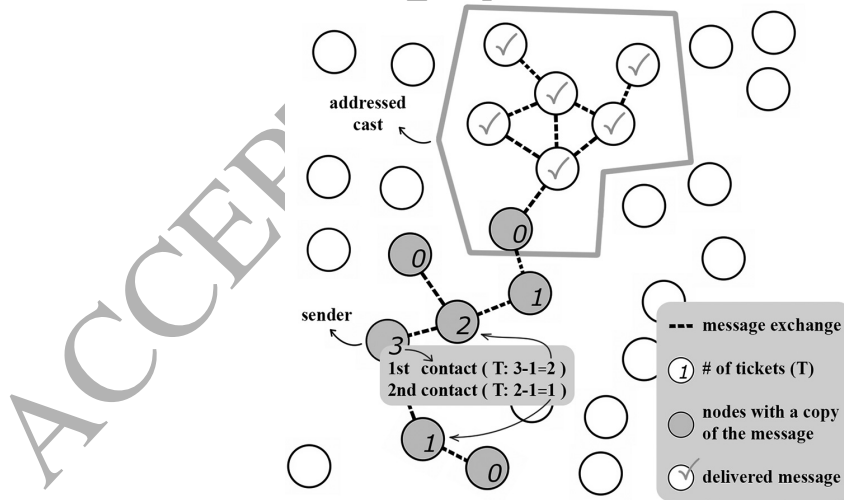


Figure 4: GSAF Routing



when these two nodes encounter other nodes in the opportunistic network. At the end of this example, a number of nodes roam outside the cast carrying a message with a  $T$  value of *zero*. These nodes will not forward the message any further. The node that resides inside the cast has also received the message ( $T$  is *zero*) but the message will keep being forwarded to recipients inside the cast (phase 2). A message can end up in the destination cast either after it was exchanged between a node outside and a node inside the cast or because it was physically carried by a node inside the cast. In both cases,  $T$  is set to 0 at the beginning of the second phase (*Line 24 of Algorithm 2*).

The value of  $T$  can be pre-specified for specific network deployments (e.g. for communication within a city) based on e.g. the expected node density and mobility patterns. In § 4.5, a sensitivity analysis for the initial value of  $T$  is presented, which indicates that values close to the best  $T$  value (with respect to the observed message delivery ratio and latency), result to very similar performance. One could therefore dynamically set  $T$ 's initial value e.g. by estimating the density of mobile devices, as in [26].

### 3.3.1. Direction-Aware GSAF (DA-GSAF)

In GSAF a device uses its location services only for determining whether it resides within a message's destination cast; no location information is exchanged among devices, therefore GSAF is privacy preserving. DA-GSAF is an extension that utilises the direction of a device to decide whether to forward a message to an encountered device or not; no mobility patterns, historical location data or planned destinations are exchanged between user devices.

DA-GSAF adds direction-awareness to GSAF's first phase (the second phase is unchanged). When a device encounters another one, other than just checking the number of remaining tickets ( $T$ ) to determine whether it can forward the message to the encountered node, also it checks if (1) it is moving away from the destination cast and (2) the encountered device is moving towards the destination cast. If these conditions are met, it passes a copy of the message to the encountered node. This way a message is forwarded only when another

**Algorithm 3** DA-GSAF Routing

---

**Let**  $C_m$  be the destination cast of message  $m$   
**Let**  $T_m$  be the # of available tickets for message  $m$   
**Let**  $B_x$  be the set of messages in node  $N_x$  buffer  
**Let**  $P_x^t$  be the coordinates of node  $N_x$  at time  $t$   
**Let**  $Dir_x$  be  $N_x$ 's direction from  $t_{prev}$  to  $t_{curr}$   
 $Dir_x = direction(P_x^{t_{prev}}, P_x^{t_{curr}})$  (Algorithm 4)  
**Let**  $Dir_x^{C_m}$  be the direction between  $N_x$  and  $C_m$   
 $Dir_x^{C_m} = direction(P_x^{t_{curr}}, C_m)$  (Algorithm 4)  
**Let**  $D(B)$  be the set of message digests from buffer  $B$

- 1: **for each** encounter with  $N_j$  **do**
- 2:   **As a sender do**
- 3:   drop expired messages from  $B_i$
- 4:   apply buffer scheduling policy
- 5:   send  $D(B_i)$  to  $N_j$
- 6:   wait for  $D(B_i \cap B_j)$
- 7:   **for each** message  $m$  in  $B_i \setminus B_j$  **do**
- 8:     **if**  $(T_m > 0)$  **then**
- 9:       **if**  $(Dir_i \neq Dir_i^{C_m})$  **then**
- 10:          send  $Dir_i^{C_m}$  to  $N_j$
- 11:          wait for  $Res \leftarrow response$
- 12:          **if**  $(Res = True)$  **then**
- 13:            forward  $m$  to  $N_j$
- 14:             $T_m \leftarrow T_m - 1$
- 15:          **end if**
- 16:       **end if**
- 17:     **else if**  $(T_m = 0)$  **then**
- 18:       **if**  $isInsideCast(N_i, C_m)$  (Algorithm 1) **then**
- 19:          forward  $m$  to  $N_j$
- 20:       **end if**
- 21:     **end if**
- 22:   **end for**
- 23:   **As a receiver do**
- 24:   wait for  $D(B_j)$  from  $N_j$
- 25:   send  $D(B_i \cap B_j)$
- 26:   **for each** received  $Dir_i^{C_m}$  **do**
- 27:     **if**  $(Dir_i = Dir_i^{C_m})$  **then**

---

---

```

28:     return response to  $N_j \leftarrow \text{True}$ 
29:   else
30:     return response to  $N_j \leftarrow \text{False}$ 
31:   end if
32: end for
33: for each received message  $m$  do
34:   if  $\text{isInsideCast}(N_i, C_m)_{(\text{Algorithm 1})}$  then
35:     deliver  $m$  to the Application Layer
36:     add  $m$  into  $B_i$ 
37:      $T_m \leftarrow 0$ 
38:   else
39:     add  $m$  into  $B_i$ 
40:      $T_m \leftarrow T_m - 1$ 
41:   end if
42: end for
43: end for

```

---

device has more potential to carry it to the destination cast. The assumption here is that even coarse-grained usage of device direction should be enough to provide better routing properties (compared to GSAF), especially in sparse networks. DA-GSAF uses four different directions in the two-dimensional space (the respective quadrants in the Cartesian coordinate system), as shown in Algorithm 4. DA-GSAF is not strictly privacy-preserving, however, direction is much less sensitive information than location history and mobility patterns, required by other approaches in the literature. Algorithm 3 depicts how DA-GSAF works. First, DA-GSAF checks if there are any tickets left to distribute a copy to an encountered device (*Line 8*). In the second condition, DA-GSAF checks if the node currently carrying the message is moving towards the direction of the destination cast ( $\text{Dir}_x$  in Algorithm 3 calculated by Algorithm 4). Moreover, the direction towards the message's destination cast is being calculated by comparing the current location to the centre of the cast ( $\text{Dir}_x^{C_m}$  in Algorithm 3 calculated by Algorithm 4). If the current node is going towards the same destination as the destination cast, the message is not forwarded (*Line 9 of Algorithm 3*). If the current node is moving away from the destination cast, DA-GSAF checks if the

**Algorithm 4** Direction Calculation**Require:**Two points with  $x$  and  $y$  coordination.**Let** Point  $A(x, y) = P_x^t$ **Let** Point  $B(x, y) = P_x^t$  or function  $castCenter(C_m)$  (Algorithm 5)

```

1: function DIRECTION( $A(x, y), B(x, y)$ )
2:   if  $((x_a = x_b) \wedge (y_a = y_b))$  then
3:     return Still
4:   else if  $(y_b \geq y_a)$  then
5:     if  $(x_b < x_a)$  then
6:       return Quadrant 2
7:     else
8:       return Quadrant 1
9:     end if
10:  else if  $(y_b < y_a)$  then
11:    if  $(x_b < x_a)$  then
12:      return Quadrant 3
13:    else
14:      return Quadrant 4
15:    end if
16:  end if
17: end function

```

**Algorithm 5** Cast Centre Calculation**Require:** $C_m$ : the destination cast of message  $m$ 

```

1: function CASTCENTRE( $C_m$ )
2:   for all defined vertices of  $C_m$  do
3:      $x_m \leftarrow mean(x_{vertex\ 1}, x_{vertex\ 2}, \dots, x_{vertex\ n})$ 
4:      $y_m \leftarrow mean(y_{vertex\ 1}, y_{vertex\ 2}, \dots, y_{vertex\ n})$ 
5:   end for
6:   return  $centrePoint(x_m, y_m)$ 
7: end function

```

encountered node is a more suitable carrier. This is done without requiring the  
 340 encountered device to reveal previous locations by letting the encountered device  
 to check whether it heads towards the destination cast. More specifically, the

node forwards the destination casts (sets of coordinates) of the messages to the encountered node and waits for the response indicating for which of these messages the encountered node is a suitable carrier (*Lines 9 to 11 of Algorithm 3*).

345 Next, the encountered node receives this information (messages' destination casts) and checks if it heads towards the same direction and responds positively or negatively (*Lines 26 to 32 of Algorithm 3*). If the response is positive, the current node forwards a copy to the encountered node (*Lines 12 to 15 of Algorithm 3*).

#### 4. Evaluation

350 In this section we present extensive experimental evaluation of our geocasting protocols. We have implemented GSAF and DA-GSAF in the *ONE* simulator [27]. We chose *ONE* because it supports (1) several realistic mobility models, (2) an extensible architecture for implementing routing protocols and sender/receiver types and (3) visualisation of both node mobility and message exchanging in real time [28]. We compare the proposed protocols to a number of existing protocols that we implemented from scratch in *ONE*. We have adapted the Epidemic[21] protocol to support geocasting, which we use as a baseline for the performance of geocasting protocols. GeoEpidemic floods messages in the network, therefore we expect maximal network overhead; flooding also affects 355 the delivery ratio given the finite size of device buffers. EVR and GeoOpp follow more sophisticated approaches for efficiently delivering messages to specific geographical areas and are the most relevant protocols to the ones proposed in this paper. We have not included other approaches, such as [7] and [16], which could be used to support geocasting, because they present significant limitations 360 compared to protocols that are specifically designed for geocasting (as described in Section 1).

##### 4.1. Measured Metrics

We evaluate our protocols by studying how the *message delivery ratio*, *delivery latency* and *network overhead* are affected when varying the number of 370 users, buffer capacity and message lifetime in various simulation scenarios.

**Message delivery ratio.** In geocasting, messages are not addressed to specific devices but to geographical areas where multiple devices may reside during the lifetime of a message. In contrast to unicast protocols where the delivery status of a message can be *delivered* or *undelivered*, in geocasting one has to take into account both the spatial and temporal aspects of cast membership. Each message has a delivery ratio, instead of a mere delivery status, which is the fraction of the number of devices that were located in the cast throughout the lifetime of the message and received the message to the total number of devices that should have received the message. The overall delivery ratio of a geocasting protocol is the ratio of the sum of the per-message delivery ratio for all created messages to the total number of created messages.

**Message delivery latency.** The same rationale is followed when measuring the message delivery latency. We measure the per-message delivery latency as the time it takes for a message to reach a device in the destination cast. The overall delivery latency is the average for all created messages.

**Network overhead.** We measure network overhead as the total number of forwarded messages in the network. For GSAF and DA-GSAF this includes messages forwarded during both phases.

In geocasting, delivery ratio and delay, can vary significantly for different casts in the network; e.g. for sparse, remote casts compared to crowded ones. In order to get a more representative view of the protocols' performance for densely populated casts, for a number of experiments we present the average (and standard deviation) for the upper 10% of the most visited casts (i.e. 2 most visited casts).

#### 4.2. Basic Simulation Setup

Below we present common parameters and settings for all simulation scenarios discussed in this section. We have experimented with different sizes of device buffers (5 to 30 MBs with a step of 5 MBs; the default being 10 MBs) and different message lifetimes (30 to 240 minutes with a step of 30 minutes; the default being 120 minutes). We have simulated all protocols with two wire-

less interfaces; (1) WiFi 802.11ac with a transmission speed of 433 Mbps and a range of 20 meters, and (2) Bluetooth 802.15 v4.0 with a transmission speed of 2 Mbps and a range of 10 meters. The simulated time for all simulations is 16 hours. The warm-up and cool-down periods for each simulation were 2 hours, therefore our results are drawn from 12 hours of simulated time. We repeat each simulation 5 times with different seeds for the mobility model. We schedule messages as follows: a sender and a destination cast are selected uniformly at random from the set of devices present in the network and the set of pre-defined casts, respectively; the message size is fixed (500KB) and a new message is scheduled every 25 to 35 seconds (a value selected uniformly at random from this time range). In all presented simulation scenarios, messages are randomly selected from the device's buffer when a device encounters another device in the network. In [29] we evaluated how different buffer scheduling policies affect GSAF's performance. The 'Highest TTL First Out' performed the best because messages that have the higher chance to reach the cast (given their large lifetime) are prioritised over messages that are likely to not make it. Due to space limitations, we do not investigate different buffer scheduling policies in this paper. Throughout the whole section we have used the same cast definition methodology and the same casts (as described in the simulation scenarios) for all studied protocols.

#### 4.3. University of Sussex Campus

In this section we evaluate the performance of the proposed protocols in scenarios where the network is within a confined geographical area. This type of opportunistic networking is very common, for example when a network is created across a festival venue or a public demonstration. We simulate networks that operate across the University of Sussex campus, which covers an area of  $1150 \times 1450$  square meters. The map was created using Google Maps and OpenJump [30]. Although GSAF and DA-GSAF do not rely on static, pre-defined casts, *for evaluation purposes only*, we have created 11 casts within the campus, as illustrated in Figure 5a. Note that defining non-overlapping

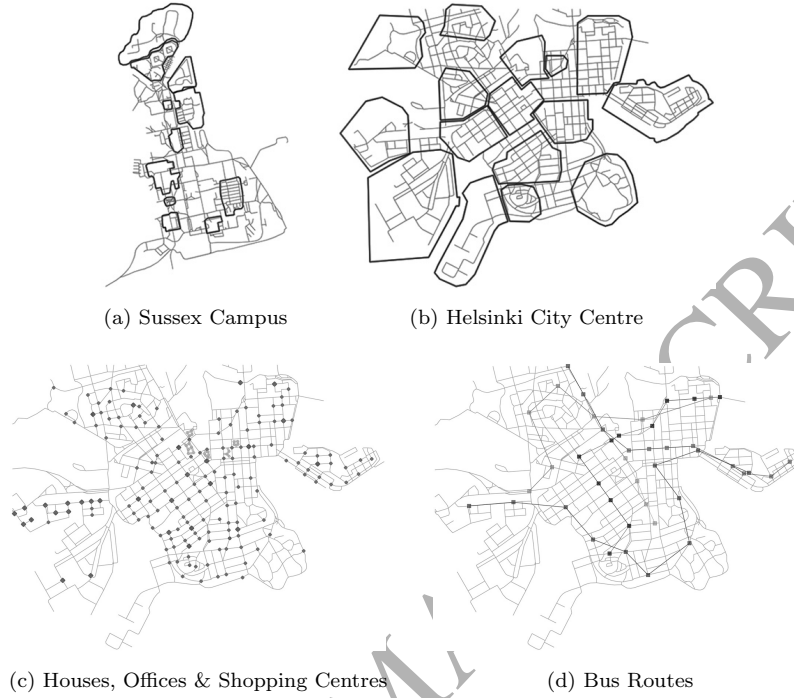


Figure 5: Simulation scenarios' maps

casts, as in Figure 5a, is not a design restriction for the proposed protocols. All messages are destined to one of these casts. As this is a university campus, in this scenario, we only simulate pedestrian users. We range the number of users that move around the campus from 40 to 320 and use ONE's *random* mobility model for all pedestrians. A user is assigned with a shortest path route from its current location in the map to a randomly selected destination point. A new such path is calculated by the simulator every time the user reaches the previously calculated destination. Unless otherwise stated, the default total number of users in our simulations is 120. Users' speed is uniformly selected from the range 0.5 - 1.5 m/s. The default initial number of tickets  $T$  is set to 3 for both GSAF and DA-GSAF protocols.



#### 4.3.1. Influence of User Density

For this set of simulations, we use the default values for the *buffer size* and *message lifetime* and vary the *number of users* in the network. In Figures 6a and 6d, we observe that the message delivery ratio increases with the number

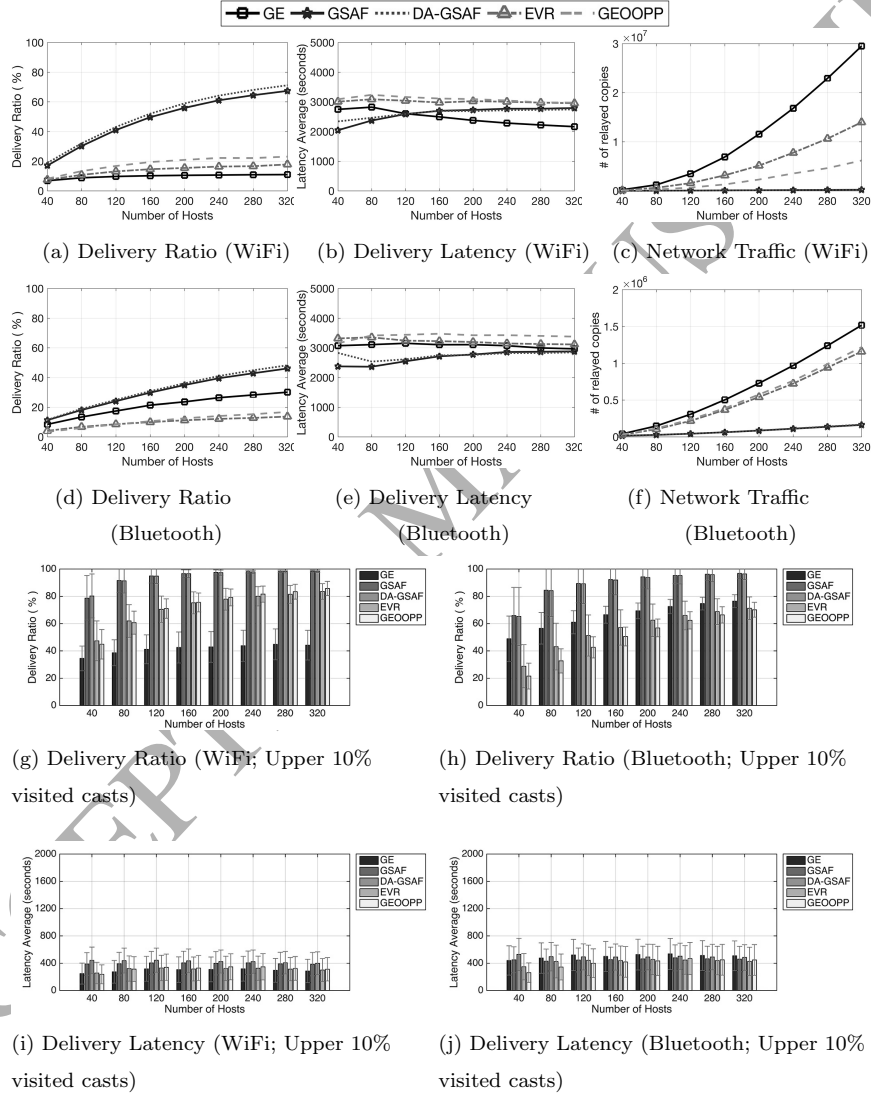


Figure 6: Influence of user density on the performance of geocasting protocols (Sussex University)

of users. GSAF and DA-GSAF deliver significantly more messages compared to all other examined protocols, while maintaining similar levels of delivery latency (although in the Bluetooth scenario GSAF and DA-GSAF perform better, as shown in Figure 6e). Figures 6c and 6f show the amount of generated network traffic in terms of relayed messages. GSAF and DA-GSAF not only outperform GeoEpidemic, EVR and GEOOPP but also do so by inducing significantly lower network overhead. Note that the observed delivery ratio for GeoEpidemic decreases when the supported transmission rate is higher (i.e. in the WiFi case). This is because when more messages are flooded in the network (as in GeoEpidemic), new messages quickly flood the network overriding older (but mostly undelivered) messages from devices' buffers.

In Figures 6g and 6h we present the results for the upper 10% of the most visited casts (bars indicate the standard deviation). The delivery ratio for both GSAF and DA-GSAF is significantly higher compared to all other considered protocols. Note that for crowded scenarios our protocols lead to delivery ratios close to 100%. GSAF and DA-GSAF perform marginally worse compared to other protocols with respect to the delivery latency (Figures 6i and 6j). We consider this as a negligible penalty we pay for significantly reducing the network overhead and increasing the delivery ratio.

#### 4.3.2. Influence of Buffer Capacity

In the second series of experiments, we keep the *user density* and *message lifetime* constant and vary the *buffer size*. In Figure 7a the delivery ratio for GSAF and DA-GSAF is higher compared to all other considered protocols. When WiFi is used, the ratio increases with the buffer availability because a device can exchange all its currently stored messages with other devices when they encounter each other. However, when Bluetooth is used (Figure 7d), the ratio reaches a plateau ( $\sim 35\%$ ) when the buffer size is 20 MBs. As the buffer size increases, more messages can be carried by each device in the network. However, there is not always enough time to exchange all buffered messages given the limited bandwidth and, more importantly, the mobility of users. As a result,

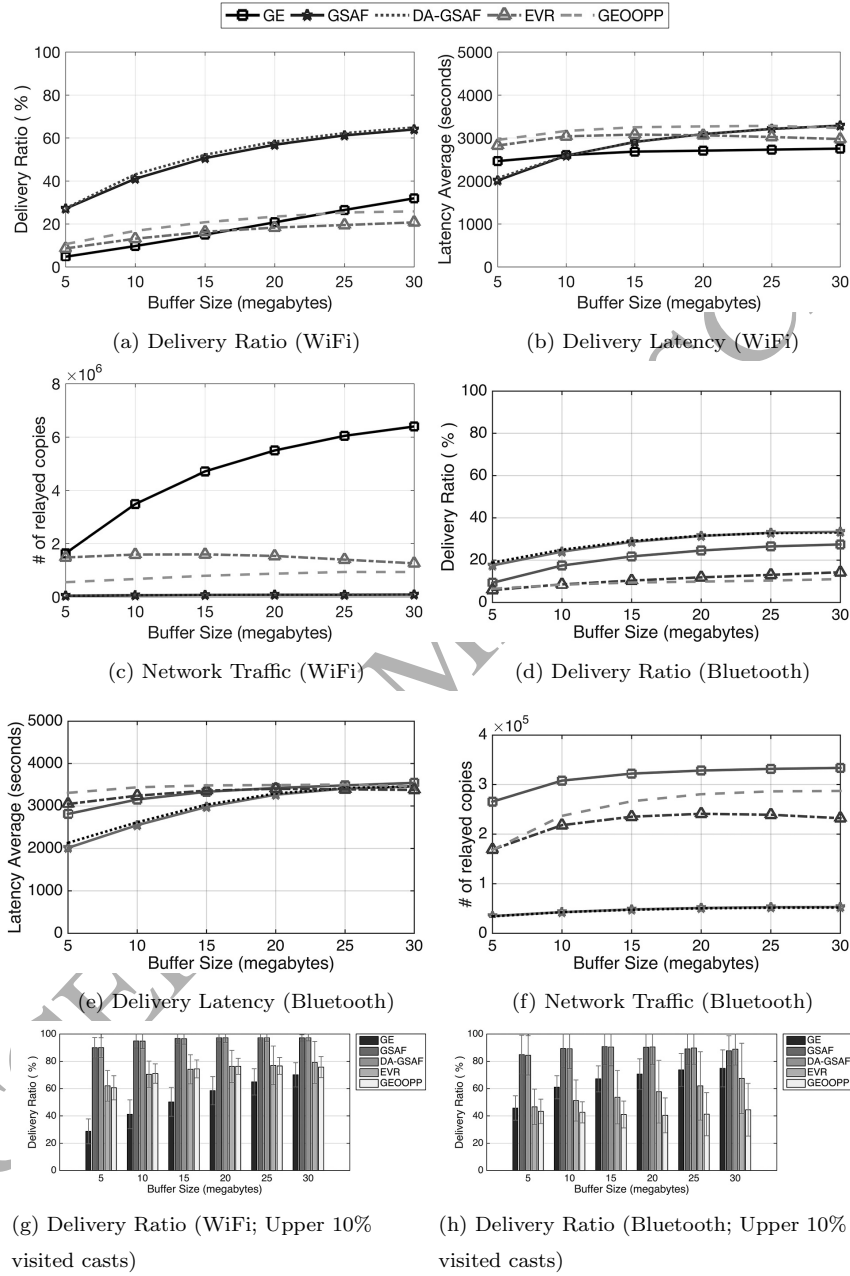


Figure 7: Influence of buffer capacity on the performance of geocasting protocols (Sussex University)

increasing the size of a buffer does not necessarily mean that the performance of a protocol is increased. Note that, as in the previous experiment, GSAF and DA-GSAF keep the induced network overhead to a minimum compared to all other considered protocols (Figures 7c and 7f). The average latency (Figures 7b and 7e) follows a similar pattern to the results presented in the previous subsection. Note that the latency increases along with the buffer size for all geocasting protocols due to the reason mentioned above. A buffer capacity of 20 to 25 MBs is adequate for GSAF and DA-GSAF (as well as for all other protocols) to handle all network traffic. In Figures 7g and 7h we observe that the proposed protocols perform exceptionally well with respect to the delivery ratio of messages in the upper 10% of the most visited casts.

#### 4.3.3. Influence of Message Lifetime

With the third set of simulations, we study the impact of message lifetime on the performance of the proposed protocols. As shown in Figure 8a, the delivery ratio decreases as the message lifetime increases. One would expect that as the lifetime of a message increases, there would be more time to deliver it in its destination cast. However, longer lifetimes imply the need for larger buffers to store (and carry) the messages and higher bandwidth to exchange them. Given the limited nature of both of them, the delivery ratio actually decreases as the message lifetime increases. Also note that longer message lifetimes mean more recipients (that resided in the cast within the message lifetime), which may have moved out of the cast and never received the message. The results for the delivery latency (Figure 8b) and network overhead (Figure 8c) show that GSAF and DA-GSAF perform better than all other considered geocasting protocols. The number of relayed messages reaches a plateau when the TTL is 90 minutes. The bottleneck here is the size of the available buffer in the network devices. The results in Figures 8d to 8e are similar to the ones presented in the previous subsection and confirm the superiority of GSAF and DA-GSAF when messages are destined to popular casts.

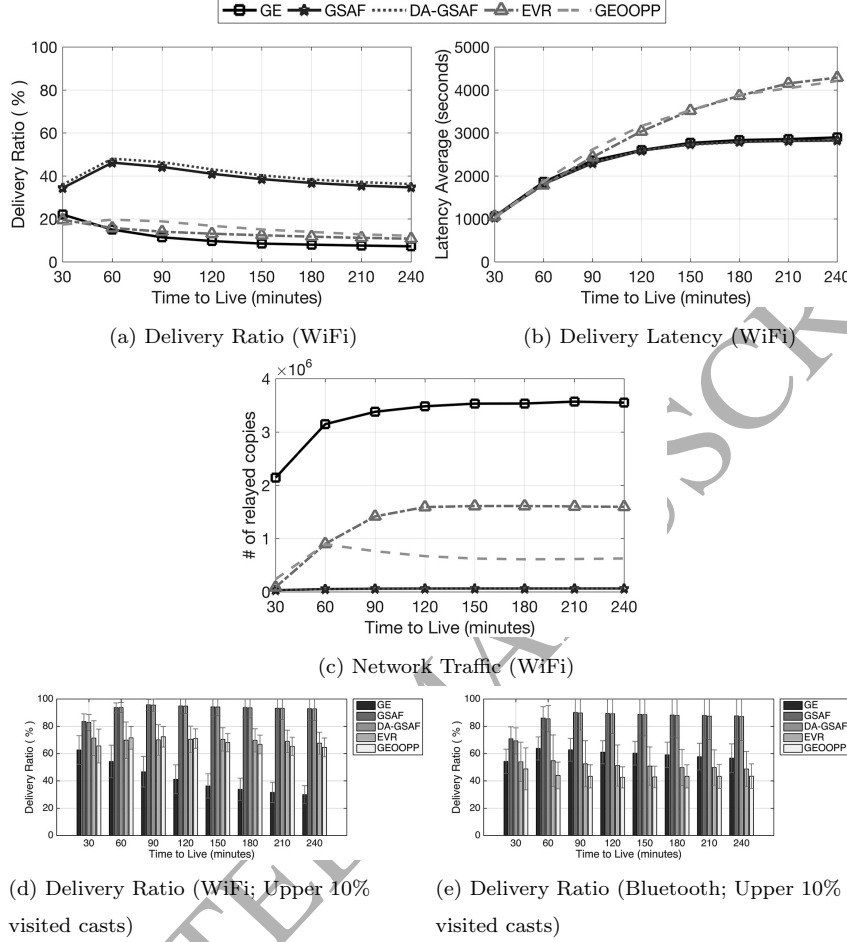


Figure 8: Influence of message lifetime on the performance of geocasting protocols (Sussex University)

#### 4.3.4. Per-Message Delivery

In order to get a better idea about how values of message delivery ratio are distributed for all different messages across all casts in the Sussex campus, we present Figures 9a to 9e, which depict scatter plots of the per-message delivery ratios (for 1400 messages being generated during each simulation) for all routing protocols. These results are extracted by running a simulation with the default values, as described in the beginning of this section. GSAF and DA-GSAF result

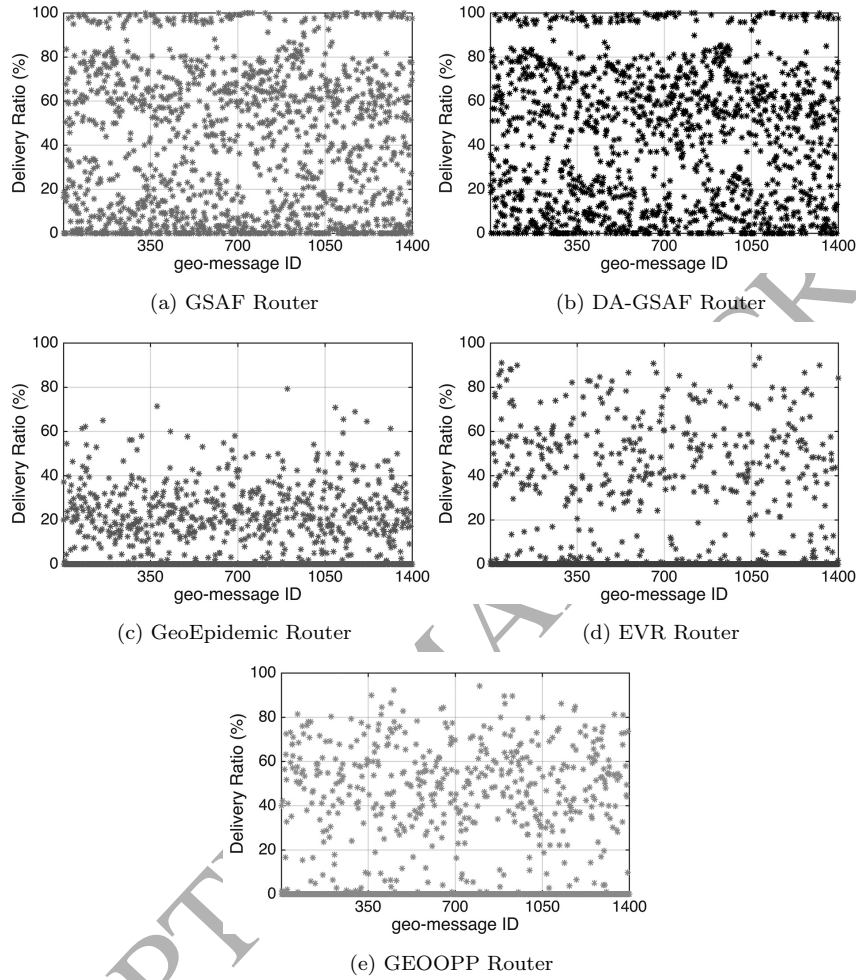


Figure 9: Per-Message Delivery Ratio (Sussex University)

in a much larger number of messages with delivery ratios that are higher than 50% compared to all other protocols. Note that the number of messages that are never delivered to their destination cast is 213 for GSAF, 145 for DA-GSAF, 848 for GeoEpidemic, 1027 for EVR and 938 for GEOOPP.

#### 4.4. Helsinki City Centre

In this section we evaluate the performance of our protocols in a simulated map of Helsinki city centre, which covers an area of  $4500 \times 3400$  square meters. Being able to perform efficiently in city-wide networking scenarios (e.g. for disaster management scenarios) is crucial for a geocasting protocol. We have defined 16 casts, as illustrated in Figure 5b and have experimented with different numbers of users (65 to 520 with step 65; the default being 195 unless otherwise stated). 60% of the users follow the working day model while 15% follow the default random mobility model. 15% of the users are driving around the city centre and 5% of the nodes are buses. The details of the working day model settings are shown in Table 1. The working day mobility model requires information about the coordinates of residential and commercial areas in the town. Figure 5c shows where homes, offices and points of interests are located in the map. There are five different bus routes in the map, each one including a number of bus stops, as illustrated in Figure 5d. Buses are distributed

Table 1: Working Day Mobility Model for Employees

<b>Speed</b>	0.8 - 1.4 $m/s$
<b>Probability to own a Car</b>	50%
<b># of Offices</b>	50
<b>Probability to go shopping after Work</b>	50%
<b>Minimum Shopping time</b>	1 Hour
<b>Maximum Shopping time</b>	2 Hours
<b># of Meeting Spots</b>	10
<b>Minimum Group size</b>	1
<b>Maximum Group size</b>	3

uniformly and serve a single route (round trip) throughout the duration of a simulation. **Note that the initial number of tickets  $T$  is set to 5 for both GSAF and DA-GSAF protocols, in this scenario.** In [29] we presented the results of simulations of GSAF, in comparison to GeoEpidemic, in the city centre of Helsinki when all user mobility was based on ONE's random mobility model. Due to space limitations and in order to provide more insights on how all studied protocols perform, we will not reproduce here the results presented in [29]. In this scenario, we evaluate the performance of GSAF, DA-GSAF, GeoEpidemic and EVR. GEOOPP [17] is computationally very heavy therefore simulations require significantly more time and memory compared to all other considered protocols. This is because GEOOPP calculates the nodes probability to carry a message towards its destination considering the node visiting every different cast (cell in [17]) in the network. This would also have implications in battery consumption in a real world deployment. Even using the Sussex HPC infrastructure, it was impossible to complete all simulations in a sensible amount of time<sup>5</sup>; we therefore decided to exclude it from the rest of the evaluation.

#### 4.4.1. Influence of User Density

As shown in Figures 10a to 10f, the overall performance of all routing protocols is worse compared to the simulation scenario (omitted here but discussed in [29]<sup>6</sup>) where the ONE's default mobility model was used. Looking at Figures 10c and 10f and the respective figures in [29], one can see that the total number of relayed messages for both the WiFi and Bluetooth scenarios is reduced by a factor of 2. This is because with many simulated users working in their offices, there are fewer opportunities to relay messages towards their destination cast. In this scenario there are casts that are rarely visited (given the distribution of the offices and shopping centres in the map) and therefore GSAF and DA-GSAF perform slightly worse there, with respect to delivering messages to these

<sup>5</sup>It would have taken around three months to run simulations for GEOOPP for all the scenarios presented below.

<sup>6</sup>A complete description of research can be found in [31]



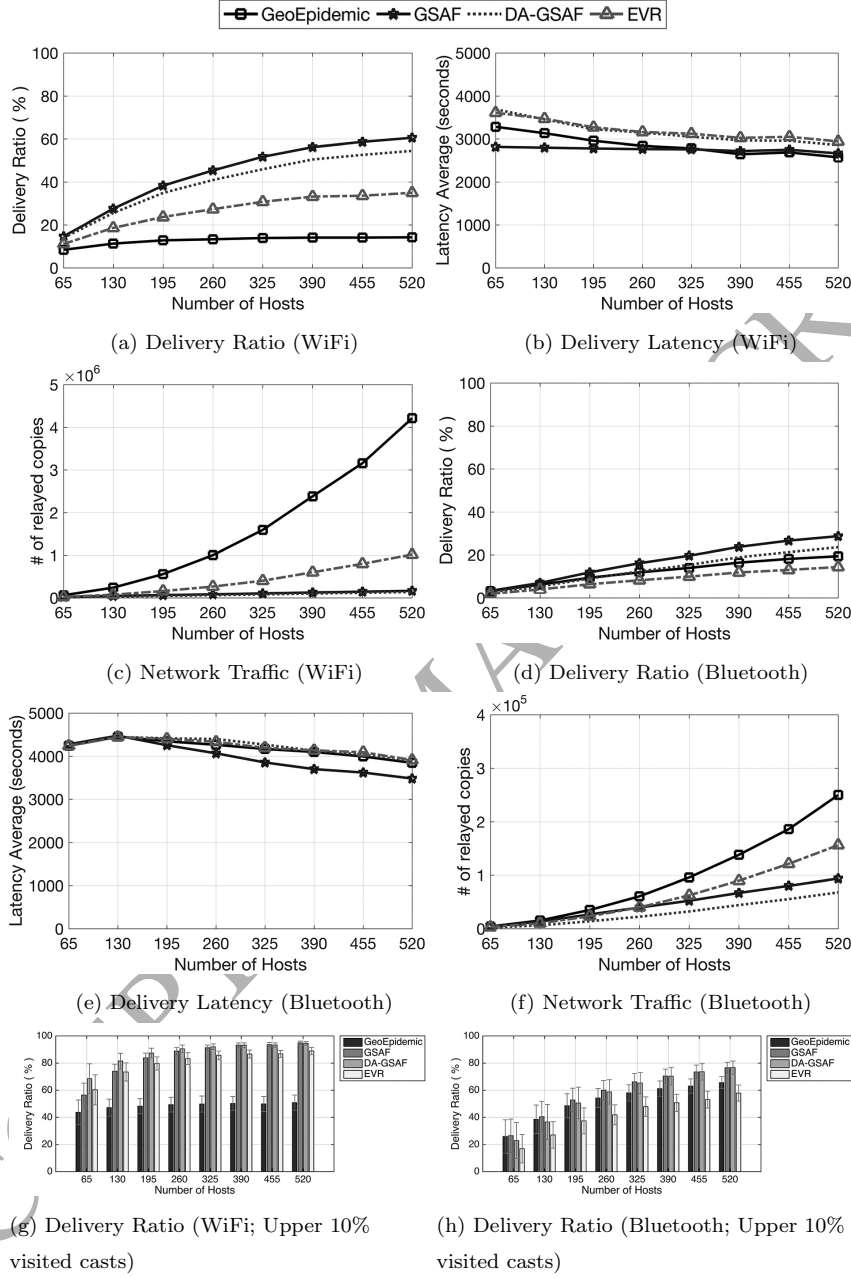


Figure 10: Influence of user density on the performance of geocasting protocols (Helsinki working day)

casts. Although EVR takes a smaller hit in performance compared to GSAF and DA-GSAF, our protocols still perform better. The results for the most visited casts are shown in Figure 10g and 10h. GSAF and DA-GSAF consistently outperform all other considered protocols.

#### 4.4.2. Influence of Buffer Capacity

The influence of buffer capacity on the performance of all studied protocols is illustrated in Figures 11a to 11e. The results are consistent to our previous

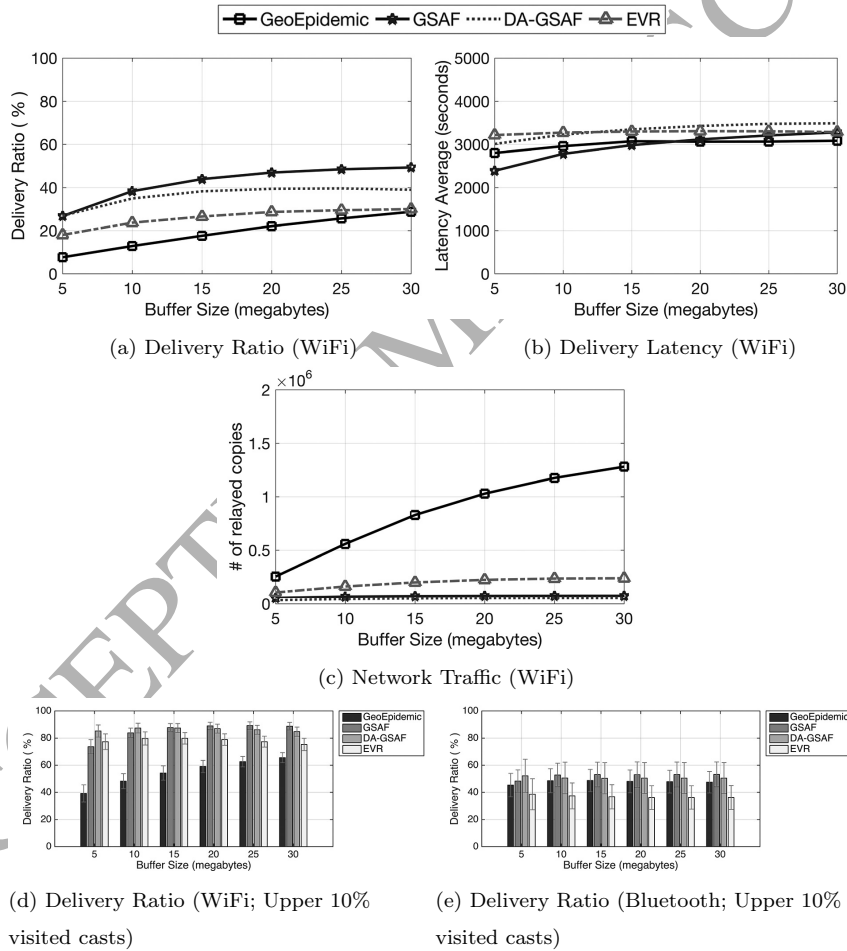


Figure 11: Influence of buffer capacity on the performance of geocasting protocols (Helsinki working day)

565 observations. Figure 11a shows that DA-GSAF's and EVR's delivery ratio is less dependent to the available buffer size, compared to the ratio of GSAF and GeoEpidemic. DA-GSAF reaches a plateau when 15 MBs of memory are available at each device for storing and carrying messages. The reason is its selection mechanism for relaying messages is such that less messages (requiring smaller  
570 buffers) are required to reach the destination cast. With respect to message latency (see Figure 11b) GSAF performs very well, while DA-GSAF is marginally worse than the other protocols. However, they both significantly outperform all other considered protocols with respect to the induced network overhead, as shown in Figure 11c. Both our protocols require around 100,000 message copies  
575 less compared to EVR. GeoEpidemic floods the network, therefore the overhead is extreme.

For the upper 10% of the visited casts in the WiFi scenario (Figure 11d), DA-GSAF outperforms all other examined protocols even for very small buffers. GSAF performs better than EVR for buffers larger than 5 MBs. Our protocols  
580 perform the best for all different buffer sizes in the Bluetooth scenario ((Figure 11e)). These observations indicate that both GSAF and DA-GSAF are suitable for networking with limited resources in terms of network bandwidth and memory.

#### 4.4.3. Influence of Message Lifetime

585 Figures 12a to 12e show the impact of message lifetime when the working day mobility model is simulated. Same as in the previous sub-sections, protocols perform worse compared to when the random mobility model is used in the simulations (as presented in [29]) because of the existence of casts that are very rarely visited. In contrast to the random mobility model where user destinations are randomly selected in the given map, with the working day model the  
590 majority of users will head to their workplaces and stay there before returning back home. GSAF and DA-GSAF outperform both EVR and GeoEpidemic for all different values of the message lifetime. As discussed earlier, the impact of message lifetime is not straightforward to understand. Longer lifetimes result

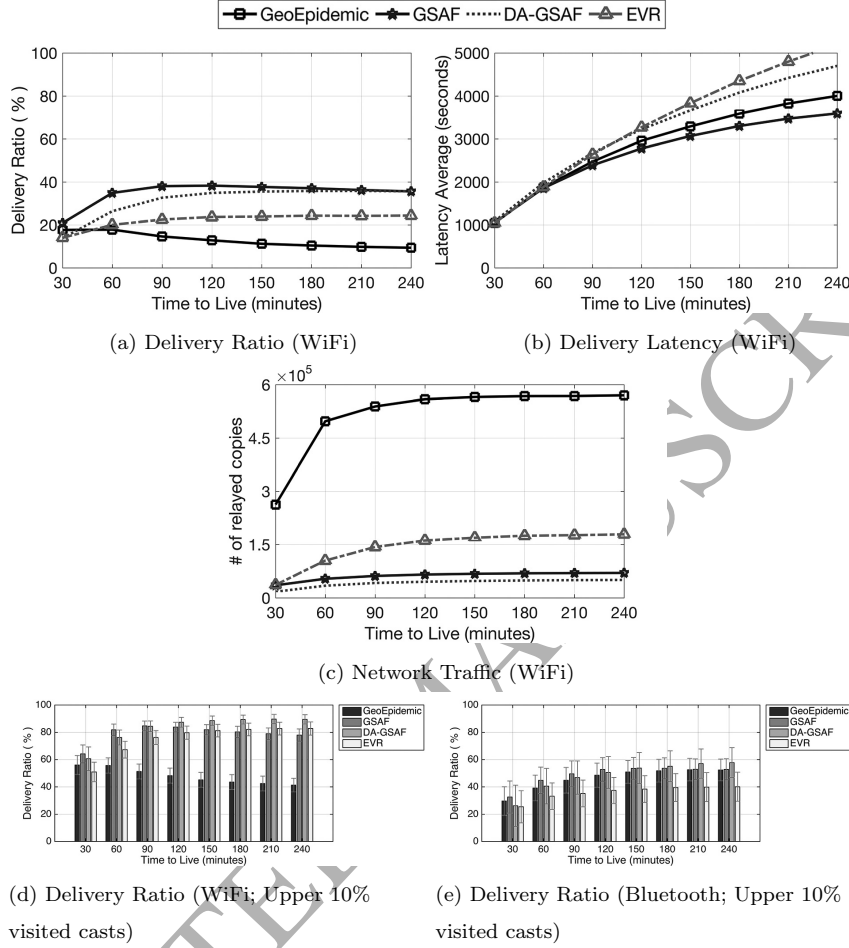


Figure 12: Influence of message lifetime on the performance of geocasting protocols (Helsinki working day)

in more messages in the network. Memory and network resources are stressed. Additionally, if the lifetime of a message is large, then a potentially very large number of users that visited its destination cast (many of which may have stayed for a short time in it) comprise the set of the message recipients. Sustaining large delivery ratios in such a case is very challenging.

#### 600 4.4.4. Per-Message Delivery

Due to lack of space we do not present the scatter plots that illustrate per-message delivery ratios. As mentioned above, because of the employed mobility model, some casts are visited very rarely. As a result, the number of undelivered messages (0% delivery ratio) increases compared to the default mobility model. 605 More specifically, the number of undelivered messages is 350 for GSAF, 510 for DA-GSAF, 850 for GeoEpidemic and 738 for EVR (out of 1400 created messages during the simulation runtime). This confirms that GSAF is performing the best while DA-GSAF comes second.

#### 4.5. Impact of Initial Ticket Value

610 The first phase of routing in GSAF and DA-GSAF heavily relies on the number of tickets ( $T$ ) a message is assigned with, upon its creation. The value of  $T$  could be dynamically adjusted (e.g. based on the inferred device density) to a value that provides the best performance. In this section we investigate how different values of  $T$  influence the performance of GSAF and DA-GSAF. 615 We are interested in looking at how sensitive our approach is to  $T$ , which, in turn, indicates what the penalty of misconfiguring  $T$  in a dynamic approach would be.

In order to study the impact of the initial ticket value to the performance of our protocols, we design a scenario based on the default scenario. We keep all 620 parameters unchanged and examine the performance of GSAF and DA-GSAF for various copy ticket values (1 to 10). The results are shown in Figures 13a to 13c and indicate that the delivery ratio is not overly sensitive to  $T$ ; therefore, in an approach where  $T$  is dynamically adjusted, missing the optimal value would not have a significant impact in the protocol's performance (i.e.  $\sim 60\%$  for value 625 '5' compared to  $\sim 53 - 58\%$  for values '4' and '6'). The results are similar for the delivery latency. Finally, the number of relayed messages increases as  $T$  increases. As shown in the figures, DA-GSAF is less sensitive to changes of  $T$  compared to GSAF. This is because DA-GSAF is more selective when it comes to passing messages to the encountered nodes. In many cases, DA-GSAF will

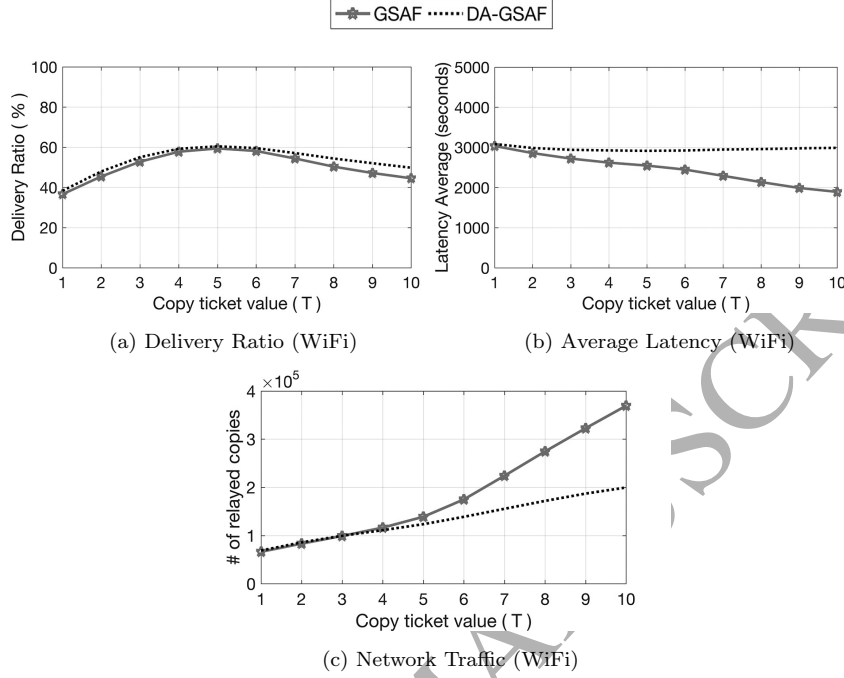


Figure 13: Influence of Copy Ticket Values on GSAF and DA-GSAF

not consume all available tickets, therefore increasing  $T$  has no effect on the behaviour of the protocol.

#### 4.6. Impact of Cast Size

With this set of experiments we examine how the cast size affects the performance of the studied protocols. For the following simulation we divide Helsinki's city centre into a variable size of equal non-overlapping rectangular casts (4  $[2 \times 2]$ , 9  $[3 \times 3]$ , 16  $[4 \times 4]$  and 25  $[5 \times 5]$ ) and use them as the recipients of messages in the simulated network. The simulation setup is the default one (with WiFi communication). Larger (and fewer) casts result in better delivery ratios and less average latencies (see Figures 14a and 14b). For large casts, messages reach the destination area quickly and then flooding takes over, as indicated by the large number of the relayed copies in Figure 14c. Moreover, large casts mean larger covered area which provides more space for nodes to

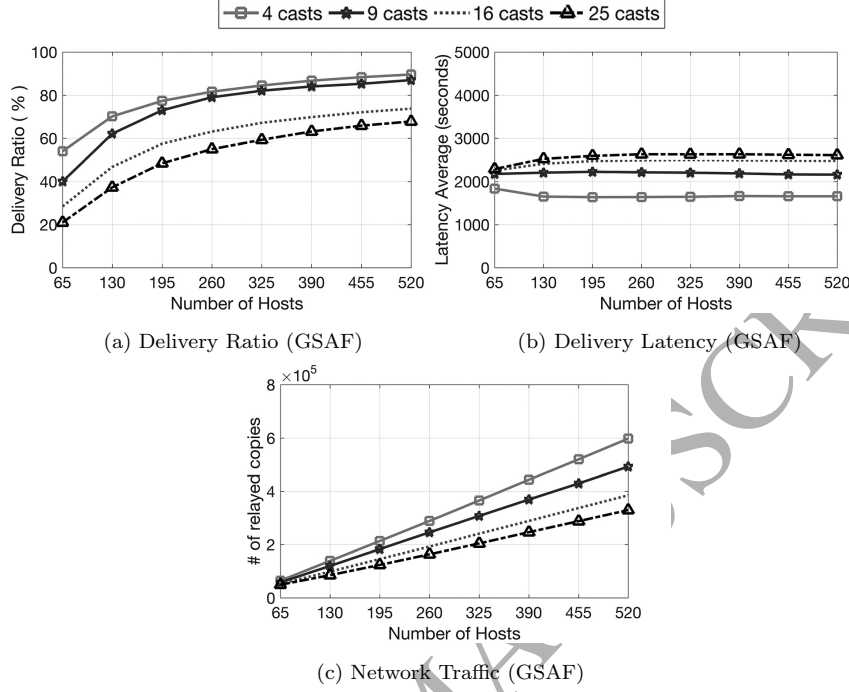


Figure 14: Impact of cast size

cover during their activity and it reduces the number of nodes who leave the cast without receiving a copy of the message.

#### 4.7. User Scalability

In this section, we investigate the behaviour of the proposed protocols for large numbers of users. Geocasting messages when a large number of users roam in the network is challenging with respect to the potentially large number of message exchanges in crowded areas due to frequent encounters between users. Messages should not become extinct very quickly due to the finite buffer space, while an adequate number of copies should be disseminated so that the destination casts can be reached. In our evaluation, we extend the simulation scenario in § 4.4 to support up to 2600 users in the network; i.e. up to five times more users than in the simulations presented above. In the following, we only

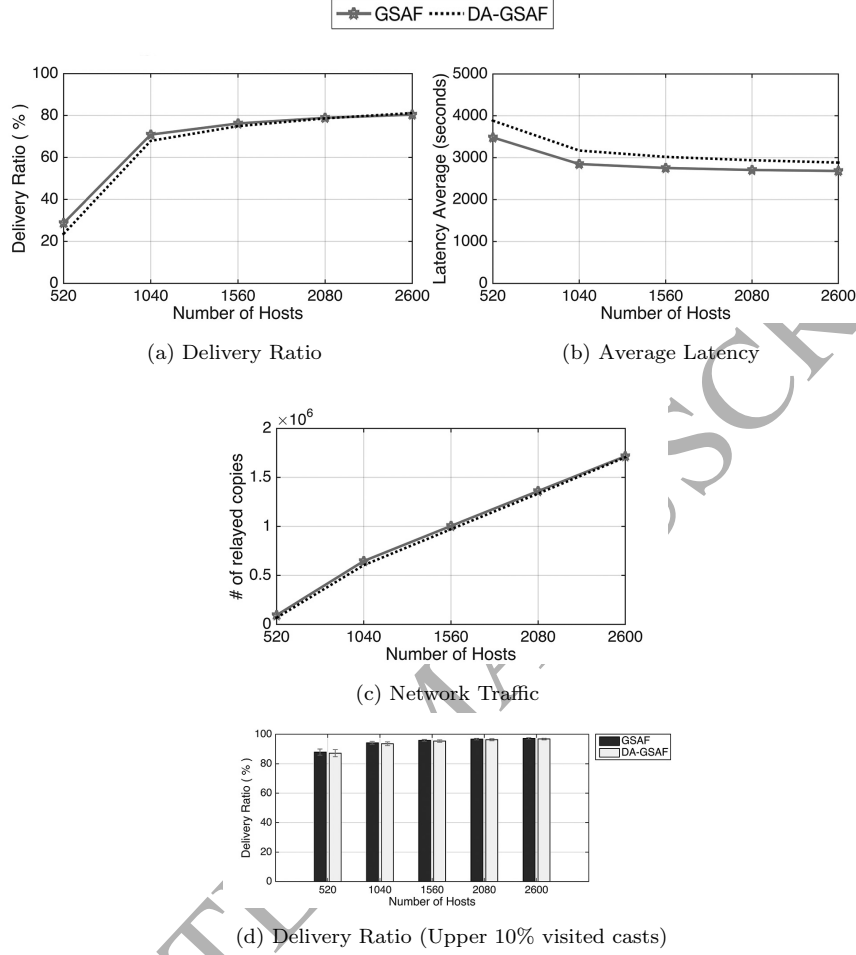


Figure 15: Influence of user density on GSAF and DA-GSAF (Crowded scenario)

present the results for Bluetooth connectivity.<sup>7</sup>

Figures 15a, 15b and 15c illustrate the delivery ratio, message latency and number of relayed messages for GSAF and DA-GSAF, respectively. The delivery ratio increases as more users are added in the network, reaching a plateau around ~80%. This seems to be a close-to-optimal value given the definition of cast

<sup>7</sup>Simulating thousands of devices with WiFi connectivity was proven extremely time consuming as more messages were being transferred during each encounter compared to the Bluetooth case.



membership; i.e. the fact that it involves a spatial and temporal aspect. More specifically, during the time it takes for a message to reach a cast, there is a number of users that entered and left the cast (and are therefore recipients of the message) without receiving the message. It would be extremely difficult, if not impossible, to deliver the message to such users. Comparing the results for 2600 against 520 nodes in Figure 15a, we observe a significant improvement in the delivery ratio as the number of users increases. There are two reasons that justify this behaviour. First, as the number of users increases, there are more opportunities for both GSAF and DA-GSAF to pass messages to encountered nodes; in previous simulations (and this one for 520 users) there were cases where not all tickets were consumed until a message reached its destination. Secondly, with more users in the network, the dissemination of messages within the destination cast is more efficient, although more messages are relayed for that purpose (see Figure 15c). Note that DA-GSAF is still slightly more selective and therefore the number of relayed messages is smaller compared to GSAF. Additionally, message sustainability within the cast increases with the number of users in the network during the message lifetime.

## 5. Conclusion

In this paper we have presented *Geocasting Spray And Flood (GSAF)*, an efficient protocol for geocasting messages in opportunistic networks. We also introduced DA-GSAF, its direction-aware extension that trades minimal sensitive information (the direction a user is heading) for better performance in sparser opportunistic network scenarios. We highlighted significant challenges in geocasting in the context of opportunistic networks and described how GSAF and DA-GSAF deal with these challenges, overcoming the limitations of existing approaches. Both protocols deliver messages to their destination casts in two phases. During the first one a message is replicated in a controlled way (using a ticketing mechanism) to encountered devices. When a message reaches its destination cast, the second phase is enabled and the message is flooded

only within the limits of the destination cast . DA-GSAF is more selective in  
 690 the way nodes exchange messages; a message is transferred to an encountered  
 device only if it heads towards the destination cast. Casts do not need to be  
 pre-defined and users are free to define their own casts even on a per-message  
 basis. Casts are polygons in a two-dimensional space, allowing for flexible and  
 efficient information dissemination.

695 We implemented GSAF and DA-GSAF in the *One Simulator* and exten-  
 sively evaluated its performance and general behaviour using a large range of  
 simulations. We have also implemented prominent geocasting protocols and  
 compared their performance to the one of our protocols. Overall, our pro-  
 tocols outperform all other considered protocols in all simulated scenarios (in  
 700 some cases the delivery latency is marginally worse for GSAF and DA-GSAF).  
 The proposed protocols are also battery- and network-friendly, requiring signif-  
 icantly less relayed message copies, compared to all other considered protocols,  
 to reach the destination casts for each message. We also presented results that  
 indicate the value of the initial number of tickets assigned to a message can  
 705 be dynamically adjusted based on e.g. user density or mobility patterns, since  
 slight mis-configuration does not significantly affect the protocols' performance.  
 Finally, we evaluated our protocols in larger scale scenarios (up to 2600 devices)  
 and with different cast sizes. The results indicate that GSAF and DA-GSAF op-  
 erate within acceptable performance limits under a diverse set of communication  
 710 scenarios.

## References

## References

- 715 [1] K. Fall, A delay-tolerant network architecture for challenged internets,  
 in: Proceedings of the SIGCOMM 2003 conference on Applications, tech-  
 nologies, architectures, and protocols for computer communications, ACM,  
 2003, pp. 27–34.

- [2] K. Fall, S. Farrell, DTN: an architectural retrospective, *IEEE Journal on Selected Areas in Communications* 26 (5) (2008) 828–836.
- [3] I. Wakeman, S. Naicken, J. Rimmer, D. Chalmers, C. Fisher, The fans  
720 united will always be connected: building a practical dtn in a football  
stadium, in: *International Conference on Ad Hoc Networks*, Springer, 2013,  
pp. 162–177.
- [4] S. Trifunovic, B. Distl, D. Schatzmann, F. Legendre, WiFi-Opp: ad-hoc-  
less opportunistic networking, in: *Proceedings of the 6th ACM workshop*  
725 *on Challenged networks*, ACM, 2011, pp. 37–42.
- [5] M. Grossglauser, D. N. C. Tse, Mobility Increases the Capacity of Ad Hoc  
Wireless Networks, *IEEE/ACM Transactions on Networking (TON)* 10 (4)  
(2002) 477–486.
- [6] V. Vukadinović, G. Karlsson, Spectral efficiency of mobility-assisted pod-  
730 casting in cellular networks, in: *Proceedings of the Second International*  
*Workshop on Mobile Opportunistic Networking*, ACM, 2010, pp. 51–57.
- [7] J. Ott, E. Hyttiä, P. Lassila, T. Vaegs, J. Kangasharju, Floating content:  
Information sharing in urban areas, in: *Pervasive Computing and Commu-*  
*nications (PerCom)*, 2011 IEEE International Conference on, IEEE, 2011,  
735 pp. 136–146.
- [8] B. Han, P. Hui, V. Kumar, M. V. Marathe, G. Pei, A. Srinivasan, Cellular  
traffic offloading through opportunistic communications: a case study, in:  
*Proceedings of the 5th ACM workshop on Challenged networks*, ACM,  
2010, pp. 31–38.
- [9] H. Luo, R. Ramjee, P. Sinha, L. E. Li, S. Lu, UCAN: a unified cellular  
740 and ad-hoc network architecture, in: *Proceedings of the 9th annual inter-*  
*national conference on Mobile computing and networking*, ACM, 2003, pp.  
353–367.

- [10] J. Erman, K. K. Ramakrishnan, Understanding the super-sized traffic of the super bowl, in: Proceedings of the 2013 conference on Internet measurement conference, ACM, 2013, pp. 353–360.
- [11] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, S. Venkataraman, J. Wang, A first look at cellular network performance during crowded events, in: Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '13, 2013, pp. 17–28.
- [12] Y. Cao, Z. Sun, Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges, IEEE Communications surveys & tutorials 15 (2) (2013) 654–677.
- [13] V. F. Mota, F. D. Cunha, D. F. Macedo, J. M. Nogueira, A. A. Loureiro, Protocols, mobility models and tools in opportunistic networks: A survey, Computer Communications 48 (2014) 5–19.
- [14] T. Wang, Y. Cao, Y. Zhou, P. Li, A survey on geographic routing protocols in delay/disruption tolerant networks, International Journal of Distributed Sensor Networks 12 (2) (2016) 3174670.
- [15] C. Maihofer, A survey of geocast routing protocols, IEEE Communications Surveys Tutorials 6 (2) (2004) 32–42.
- [16] M. Piorkowski, Mobility-centric geocasting for mobile partitioned networks, in: Network Protocols, 2008. ICNP 2008. IEEE International Conference on, IEEE, 2008, pp. 228–237.
- [17] S. Lu, Y. Liu, GeoOpp: Geocasting for opportunistic networks, in: Wireless Communications and Networking Conference (WCNC), 2014 IEEE, IEEE, 2014, pp. 2582–2587.
- [18] Y. Ma, A. Jamalipour, Opportunistic geocast in disruption-tolerant networks, in: Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE, IEEE, 2011, pp. 1–5.

- [19] Y. Ma, A. Jamalipour, Opportunistic geocast in large scale intermittently connected mobile ad hoc networks, in: Communications (APCC), 2011 17th Asia-Pacific Conference on, IEEE, 2011, pp. 445–449.
- [20] T. Spyropoulos, K. Psounis, C. S. Raghavendra, Spray and wait: an efficient routing scheme for intermittently connected mobile networks, in: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, ACM, 2005, pp. 252–259.
- [21] A. Vahdat, D. Becker, Epidemic routing for partially connected ad hoc networks, Tech. rep., Technical Report CS-200006, Duke University (2000).
- [22] R. Harle, A survey of indoor inertial positioning systems for pedestrians, IEEE Communications Surveys Tutorials 15 (3) (2013) 1281–1293.
- [23] I. Parris, T. Henderson, The impact of location privacy on opportunistic networks, in: World of Wireless, Mobile and Multimedia Networks (WoW-MoM), 2011 IEEE International Symposium on a, IEEE, 2011, pp. 1–6.
- [24] E. Haines, Point in polygon strategies, Graphics gems IV 994 (1994) 24–26.
- [25] W. Zhao, M. Ammar, E. Zegura, Multicasting in delay tolerant networks: semantic models and routing algorithms, in: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, ACM, 2005, pp. 268–275.
- [26] J. Weppner, P. Lukowicz, Bluetooth based collaborative crowd density estimation with mobile phones, in: Pervasive computing and communications (PerCom), 2013 IEEE international conference on, IEEE, 2013, pp. 193–200.
- [27] A. Keränen, J. Ott, T. Kärkkäinen, The ONE simulator for DTN protocol evaluation, in: Proceedings of the 2nd international conference on simulation tools and techniques, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, p. 55.

- [28] The Opportunistic Network Environment simulator, <http://www.netlab.tkk.fi/tutkimus/dtn/theone/>.
- 800 [29] A. Rajaei, D. Chalmers, I. Wakeman, G. Parisi, Gsaf: Efficient and flexible geocasting for opportunistic networks, in: World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2016 IEEE 17th International Symposium on A, IEEE, 2016, pp. 1–9.
- [30] OpenJump open source Geographic Information System (GIS), <http://www.openjump.org/>.
- 805 [31] A. Rajaei, Efficient and flexible geocasting for opportunistic networks, Ph.D. thesis, University of Sussex (December 2016).  
URL <http://sro.sussex.ac.uk/66039/>